A SYSTEM AND METHOD FOR THE USE OF RESET LOGIC IN HIGH AVAILABILITY SYSTEMS

Field of the Invention

The illustrative embodiment of the present invention relates generally to high availability systems and more particularly to the use of reset logic in high availability systems.

Background of the Invention

The term "high availability system" is used in the telecommunications industry to specify a system meeting an availability requirement of 99.999%. Frequently, the system availability requirement is met through the use of multiple redundant components which may include multiple processors. Typically, processors possess three activation states; active, standby and reset. A processor in an active state is fully functioning and processing all of its assigned operations. A processor in a standby state is usually a redundant processor which is available to replicate the activities of an active processor if needed. A processor in a standby state performs only a small percentage of the operations of which it is capable. For example, both an active processor and a standby processor may receive the same information from a system component, but ordinarily only the active processor will act on the information while the standby processor merely monitors the information. However, there are exceptions to the general rule that only the active processor acts on received information. A processor in a reset state is frozen such that it is treated as functionally removed from the system by the other system components. In the event that one of the processors in the system fails, the failure of the processor must be detected and the processor put in a reset state so that it can cause no other failures in the system.

A number of techniques have been devised to detect and reset failed processors in a high availability system. In one technique, a hardware watchdog timer periodically verifies that a processor is capable of performing a defined task in a pre-determined amount of time. If the watchdog timer determines that a processor has failed to perform the defined task in the pre-determined amount of time, the timer generates a reset signal to the processor. This technique is deficient for a couple of different reasons. The defined task is usually a very simple task that is not indicative of the requirements for the processor, such as performing a simple math operation in a given amount of time.

Also, the watchdog timer is built into the hardware which limits its adaptability for different tests. A second technique that has been used in high availability systems is the use of a second processor to determine whether a first processor has failed. In the event the second processor determines the first processor has failed, the second processor generates a reset signal to the first processor. The problem associated with this technique is that it raises the question of who is watching the watcher. In other words, if the second processor fails rather than the first processor, the second processor's failure may be undetected and the first processor may be reset erroneously.

Summary of the Invention

The illustrative embodiment of the present invention provides a reliable means of detecting the failure of a processor in a high availability system, and a method of resetting the failed processor. The term high availability system, as used herein, refers to a system which is immune to the failure of any single active component. Through the use of probability theory and multiple redundant components, the illustrative embodiment of the present invention provides a means for detecting, verifying and resetting a failed processor while meeting the system availablity requirements of a high availability system.

In one embodiment of the present invention, a computer system, such as a high availability computer system, includes an active system control processor with a plurality of activation states. The high availability computer system also includes a standby system control processor with a plurality of activation states as well as a plurality of system components. The system control processors control a plurality of other processors, each with their own activation state, in the system. When an error is detected by the active system control processor and verified by the standby system control processor, both the active and standby system control processors send a reset signal to the other processor with the error. The activation state for the other processor is changed to a reset activation state, an action that functionally removes it from the system.

In another embodiment of the present invention, a computer system, such as a high availability computer system, includes both an active and standby system control processor with a plurality of activation states. The high availability computer system also includes a plurality of other processors and components, including a system component with a reset function capable of changing the activation state of an system

control processor. When an error in one of the other processors is detected by the active system control processor, the standby system control processor attempts to verify the error. If the detected error is not verified by the standby system control processor the other processor is not reset. Based on probability theory, the active system control processor is treated as having erroneously detected the initial error. Accordingly, instead of resetting the other processor, the active system control processor is reset. The standby system control processor and the system component with a reset function jointly send a reset signal to the active system control processor which changes its activation state from an active state to a reset state.

In an alternate embodiment of the present invention, a computer system, such as a high availability computer system, includes an active and a standby system control processor, both with a plurality of activation states, and a plurality of other processors. Additionally, the system includes a system component with a reset function capable of changing the activation state of a system control processor. When an error is detected in the active system control processor by the standby system control processor, the system component with the reset function is notified and attempts to verify the error in the active system control processor. If the error in the active system control processor is verified, the system component with the reset function and the standby system control processor jointly send a reset signal to the active system control processor which changes its activation state from an active state to a reset state.

In another embodiment of the present invention, a computer system, such as a high availability computer system, includes an active and a standby system control processor, both with a plurality of activation states, and a plurality of other processors. Additionally, the system includes a system component with a reset function capable of changing the activation state of a system control processor. When an error is detected in the active system control processor by the standby system control processor, the system component with the reset function is notified and attempts to verify the error in the active system control processor. If the error in the active system control processor is not verified, the active system control processor is not reset. Based on probability theory, the standby system control processor is treated as having erroneously detected the initial error. Accordingly, instead of resetting the active system control processor, the standby system control processor is reset. The active system control processor and the system component with a reset function jointly send a reset signal to the standby system control processor which changes its activation state from an active state to a reset state.

Brief Description of the Drawings

Figure 1 is a block diagram depicting a high availability computer system environment suitable for practicing an embodiment of the present invention;

Figure 2 is a flow chart of the sequence of steps executed by an illustrative embodiment of the present invention resetting another processor in the system following detection of an error in the other processor by the active system control processor;

Figure 3 is a flow chart of the sequence of steps executed in an embodiment of the present invention when the standby system control processor detects an error in the active system control processor; and

Figure 4 is a flow chart of the sequence of steps executed by an embodiment of the present invention when a system component self-detects an error and notifies the active system control processor.

Detailed Description

The illustrative embodiments of the present invention provide a mechanism to accurately and quickly detect processor and component failure, including placing a failed processor into a reset activation state where it is functionally removed from the system. Many systems are required to perform continuously even in the event of component failure. High availability systems, such as those utilized in the telecommunications industry, typically must recover from failures in system components within 50 milliseconds or less. As noted above, this is often achieved using multiple redundant components including multiple processors. Having the available hardware to recover from system failure is only one part of the failure recovery process, however. It is also necessary to accurately detect processor and component failures and switch to the backup redundant processor and/or components in a timely manner. The illustrated embodiments of the present invention enable the diagnosis and recovery from processor problems in the system with minimal service disruptions.

Figure 1 depicts a block diagram of a high availability computer system suitable for practicing an illustrative embodiment of the present invention. The high availability computer system 1 maybe a networking switch, such as the SN4000 from Sycamore Networks of Chelmsford, Massachusetts. The high availability computer system includes a system control processor (SCP) in an active activation state (active system control processor) 2 and a system control processor in a standby activation state (standby system control processor) 3. The high availability computer system 1 also

includes a system component 4 with a reset function in an active activation state and may include a system component with a reset function in a standby state 5. Also included in the high availability computer system 1 are a plurality of other processors (OPs) 6, 7, 8, and 9, which are subject to the commands of the active system control processor. The system control processors, other processors, and system components with reset functions may appear on I/O cards 10 interconnected by way of a connection media 7. The connection media 7 may be a mid-plane or backplane in a networking switch. Also located on the I/O cards 10 may be software 11 which performs self diagnostic tests which check the status of the I/O card and its processor 7.

Figure 2 is a flow chart of the sequence of steps followed by an illustrative embodiment of the present invention following the active system control processor 2 detecting an error in another processor 6, 7, 8 and 9. The active system control processor 2, periodically checks the operating status of the other processors 6, 7, 8, and 9 present in the high availability computer system 1 (step 16). For example, the active system control processor 2 may periodically check status at a set time. Alternatively, the active system control processor 2 may check status during periods of peak processor usage or periods of low processor usage. The active system control processor 2 may "ping" the other processors 6, 7, 8, and 9 to detect their working status (eg.: send a message requiring a response), or may perform some other more substantive tests to determine errors in the other processors (step 18). The period of time in which the active system control processor 2 is querying the other processors 6, 7, 8, and 9 to determine their working status is very small, on the order of milliseconds. If no error is detected in the other processors 6, 7, 8, and 9, the active system control processor 2 waits a pre-determined amount of time and then checks with the other processors again in a continuous cycle (step 16). If the active system control processor 2 detects an error in one of the other processors 6, 7, 8, and 9, the standby system control processor 3 is notified and attempts to verify the error in the identified other processor (step 20). The standby system control processor 3 first makes a determination as to whether or not the identified other processor 6, 7, 8, and 9 is malfunctioning (step 22). If the standby system control processor 3 determines that the identified other processor 6, 7, 8, and 9 is not malfunctioning, the standby system control processor asserts a reset command to the active system control processor 2
(step 24).

The decision to assert a reset command to the active system control processor 2 rather than the identified other processor 6, 7, 8, and 9 is based on probability theory.

The probability theory is based on the fact that the period of time between the active system control processor 2 detecting an error and the standby system control processor 3 querying the other processors 6, 7,8, and 9 to verify the error is very small, on the order of milliseconds. If the active system control processor identifies an error in one of the other processors and the standby system control processor 3 does not detect the same error, one of two scenarios is possible. Either the active system control processor 2 is correct and, thus, both the identified other processor 6, 7, 8, and 9 and the standby system control processor 3 are malfunctioning, or the active system control processor 2 erroneously detected the error. Since the probability of two processors failing within the same small time period is extremely remote, the active system control processor 2 is deemed to have failed and the standby system control processor 3 sends a reset signal to the active system control processor to change its activation state to a reset activation state (step 24). Once the active system control processor 2 assumes a reset activation state, the active system control processor is functionally removed from the system and can cause no further damage to the working conditions of the system. Following asserting the reset command to the active system control processor 2, the activation state of the standby system control processor 3 is changed from standby to active and the standby system control processor becomes the active system control processor (step 26). If the high availability computer system 1 includes a third system control processor, the third system control processor replaces the original standby control processor 3 as the standby system control processor.

If the standby system control processor 3 verifies the error in the other processor 6, 7, 8, and 9 that was originally identified by the active system control processor 2, the active and standby system control processors 2 and 3 jointly assert a reset command to the identified other processor (step 28). The identified other processor 6, 7, 8, and 9 transitions from an active activation state to a reset activation state, an activation state which functionally removes the identified processor from the system. A redundant processor that duplicates the functions of the reset processor is switched to an active state to replace the reset processor. (step 30).

Figure 3 is a flow chart of the sequence of steps followed by an illustrative embodiment of the present invention when a standby system control processor 3 performing routine checks on the working status of the active system control processor 2 detects an error in the active system control processor. The sequence of steps begins when the standby system control processor 3 detects an error in the active system control processor 2 (step 34). The standby system control processor 3 sends a

notification of the identified error in the active system control processor 2 to a system component with a reset function 4 (step 36). The system component with a reset function 4 then attempts to verify the error identified by the standby system control processor 3 (step 38). The system component with a reset function 4 performs pre-determined tests on the active system control processor 2 in an attempt to verify the identified error (step 40). If the identified error is not verified by the system component with a reset function 4, the system component with the reset function asserts a reset command to the standby system control processor 3 which incorrectly identified the original error. The rationale behind resetting the standby system control processor 3 is the same as noted above. Either both the active system control processor 2 and the system component with a reset function 4 are malfunctioning, or the standby system control processor 3 incorrectly identified a malfunction in the active system control processor 2. Based on the probability of two errors occurring in the extremely small amount of time in which the verification process takes place, the standby system control processor 3 is placed in a reset activation state for having incorrectly identified a malfunction in the active system control processor 2 (step 44). If another system control processor is available, it replaces the reset standby system control processor 3.

If the system component with a reset function 4 verifies the error identified by the standby system control processor 3, the standby system control processor 3 and the system component with the reset function 4 jointly assert a reset command to the active system control processor 2 (step 46). The active system control processor 2 activation state is changed from active to reset, thereby functionally removing it from the system, and the activation state of the standby system control processor 3 is switched from standby to active making it the active control processor in the high availability computer system 1. If another system control processor is available, it replaces the activated standby system control processor 3 as the standby system control processor.

In some embodiments of the present invention, the processors and system components of the high availability computer system 1 are located on cards such as I/O cards 10. Also located on the I/O cards 10, is software 11 capable of performing maintenance checks on both the card and processor or system component located on the card. **Figure 4** depicts a sequence of events followed by an illustrated embodiment of the present invention which occurs when software located on an I/O card identifies a malfunction in either the I/O card 10 or the processor or system component located on the I/O card. The software located on the I/O card 10 periodically performs maintenance checks of the working condition of the I/O card and any processor or

system component located on the I/O card. When the software on the I/O card 10 locates or detects an error (step 52), the software sends notification of the detected error to the active system control processor 2(step 54). The active system control processor 2 thereafter sends a reset message to the other processor 6, 7, 8, and 9 or system component 4, 5 on the I/O card with the malfunction (step 56). In some embodiments, the active system control processor first verifies the identified malfunction. After sending a reset message to the identified other processor or system component, the active system control processor 2 sends a message activating a redundant processor or system component to take the place of the reset processor or system component (step 58). Those skilled in the art will recognize that while the figures and descriptions herein have been discussed with reference to computer systems containing two system control processors, the method of the present invention is equally applicable to high availability systems and other systems possessing more than two system control processors and more than two system components with reset functions.

The illustrative embodiments discussed herein have used two system control processors or a system control processor in combination with a system component possessing a reset function to assert reset signals to a malfunctioning processor or component. Those skilled in the art will recognize that in alternate embodiments of the present invention a single system control processor or system component with a reset function may be used to assert a reset signal to a malfunctioning processor or system component. Likewise, other embodiments using combinations of three or more system control processors and/or system components with reset functions to assert reset signals to a malfunctioning processor or component are possible within the scope of the present invention.

It will thus be seen that the invention attains the objects made apparent from the preceding description. Since certain changes may be made without departing from the scope of the present invention, it is intended that all matter contained in the above description or shown in the accompanying drawings be interpreted as illustrative and not in a literal sense. Practitioners of the art will realize that the system configurations depicted and described herein are examples of multiple possible system configurations that fall within the scope of the current invention. Likewise, the sequence of steps utilized in the illustrated flowcharts are examples and not the exclusive sequence of steps possible within the scope of the present invention.